


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
 The ACM Digital Library The Guide

reaching set

SEARCH

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)
Terms used **reaching set**Found **98,424** of **160,172**

Sort results by

relevance Save results to a Binder[Try an Advanced Search](#)

Display results

expanded form Search Tips[Try this search in The ACM Guide](#) Open results in a new window

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale **1** [On the computation of the set of reachable states of hybrid models](#)

A. S. Krishnakumar, Kwang-Ting Cheng

June 1994 **Proceedings of the 31st annual conference on Design automation**Full text available: [pdf\(252.04 KB\)](#) Additional Information: [full citation](#), [references](#), [index terms](#)**2** [New scan-based test techniques: On the generation of scan-based test sets with reachable states for testing under functional operation conditions](#)

Irith Pomeranz

June 2004 **Proceedings of the 41st annual conference on Design automation**Full text available: [pdf\(120.42 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Design-for-testability (*DFT*) for synchronous sequential circuits allows the generation and application of tests that rely on non-functional operation of the circuit. This can result in unnecessary yield loss due to the detection of faults that do not affect normal circuit operation. Considering single stuck-at faults in full-scan circuits, a test vector consists of a primary input vector *U* and a state *S*. We say that the test vector consisting of *U* and *S* relies ...

Keywords: functional tests, reachable states, scan design**3** [Reconstructing a history of recombinations from a set of sequences](#)

John Kececioglu, Dan Gusfield

January 1994 **Proceedings of the fifth annual ACM-SIAM symposium on Discrete algorithms**Full text available: [pdf\(1.09 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**Keywords:** bottleneck optimality, computational biology, directed hypergraphs, edit distance, evolutionary trees, recombination**4** [Interconvertibility of set constraints and context-free language reachability](#)

David Melski, Thomas Reps

December 1997 **ACM SIGPLAN Notices, Proceedings of the 1997 ACM SIGPLAN**

symposium on Partial evaluation and semantics-based program manipulation, Volume 32 Issue 12

Full text available:  [pdf\(1.56 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We show the interconvertibility of context-free-language reachability problems and a class of set-constraint problems: given a context-free-language reachability problem, we show how to construct a set-constraint problem whose answer gives a solution to the reachability problem; given a set-constraint problem, we show how to construct a context-free-language reachability problem whose answer gives a solution to the set-constraint problem. The interconvertibility of these two formalisms offers an ...

5 Set consensus using arbitrary objects (preliminary version) 

Maurice Herlihy, Sergio Rajsbaum

August 1994 **Proceedings of the thirteenth annual ACM symposium on Principles of distributed computing**

Full text available:  [pdf\(955.72 KB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

6 Agreement is harder than consensus: set consensus problems in totally asynchronous systems 

Soma Chaudhuri

August 1990 **Proceedings of the ninth annual ACM symposium on Principles of distributed computing**

Full text available:  [pdf\(1.43 MB\)](#)

Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

7 Electronic commerce technologies (ECT): An approach to handle real time and probabilistic behaviors in e-commerce: validating the SET protocol 

G. Diaz, K. Larsen, J. Pardo, F. Cuartero, V. Valero

March 2005 **Proceedings of the 2005 ACM symposium on Applied computing**

Full text available:  [pdf\(196.43 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this work we describe an approach to deal with systems having at the same time probabilistic and real-time behaviors. The main goal in the paper is to show the automatic translation from a real time model based on UPPAAL tool, which makes automatic verification of Real Time Systems, to the RAPTURE tool, which makes verification of probabilistic systems. Furthermore, this approach allows us to use the best techniques developed in both tools (abstraction, refinement, state space reduction, etc). ...

Keywords: authentication protocols, authentication protocols and model checking, e-commerce, security, system verification and model checking

8 On the improbability of reaching Byzantine agreements 

R. L. Graham, A. C. Yao

February 1989 **Proceedings of the twenty-first annual ACM symposium on Theory of computing**

Full text available:  [pdf\(954.92 KB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

It is well known that for the Byzantine Generals Problem, no deterministic protocol can exist for an n -processor system if the number t of faulty processors is allowed to be as large as $n/3$. In this paper we investigate the maximum achievable agreement probability α in a model in which the faulty processors can be as devious and powerful as possible. We also

discuss a restricted model with &Bgr;< ...

9 Set Manipulation with Boolean Functional Vectors for Symbolic Reachability Analysis

Amit Goel, Randal E. Bryant

March 2003 **Proceedings of the conference on Design, Automation and Test in Europe - Volume 1 DATE '03**

Full text available:  pdf(144.26 KB)



Additional Information: [full citation](#), [abstract](#), [index terms](#)

Symbolic techniques usually use characteristic functions for representing sets of states. Boolean functional vectors provide an alternate set representation which is suitable for symbolic simulation. Their use in symbolic reachability analysis and model checking is limited, however, by the lack of algorithms for performing set operations. We present algorithms for set union, intersection and quantification that work with a canonical Boolean functional vector representation and show how this enab ...

10 Algorithms for approximate FSM traversal

Hyunwoo Cho, Gary D. Hachtel, Enrico Macii, Bernard Plessier, Fabio Somenzi

July 1993 **Proceedings of the 30th international conference on Design automation**

Full text available:  pdf(827.73 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

11 Data flow analysis of communicating finite state machines

Wuxu Peng, S. Puroshothaman

July 1991 **ACM Transactions on Programming Languages and Systems (TOPLAS)**

Volume 13 Issue 3

Full text available:  pdf(2.43 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: communicating finite state machines, static analysis

12 A formal protection model of security in centralized, parallel, and distributed systems

Glenn S. Benson, Ian F. Akyildiz, William F. Appelbe

August 1990 **ACM Transactions on Computer Systems (TOCS)**, Volume 8 Issue 3

Full text available:  pdf(2.17 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

One way to show that a system is not secure is to demonstrate that a malicious or mistake-prone user or program can break security by causing the system to reach a nonsecure state. A fundamental aspect of a security model is a proof that validates that every state reachable from a secure initial state is secure. A sequential security model assumes that every command that acts as a state transition executes sequentially, while a concurrent security model assumes that multiple commands execut ...

Keywords: access control, concurrency control, distributed system security, operating system security, protection model

13 Lazy abstraction

Thomas A. Henzinger, Ranjit Jhala, Rupak Majumdar, Grégoire Sutre

January 2002 **ACM SIGPLAN Notices , Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 37 Issue 1

Full text available:  pdf(2.05 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

One approach to model checking software is based on the *abstract-check-refine* paradigm: build an abstract model, then check the desired property, and if the check fails, refine the model and start over. We introduce the concept of *lazy abstraction* to integrate and optimize the three phases of the abstract-check-refine loop. Lazy abstraction continuously builds and refines a single abstract model on demand, driven by the model checker, so that different parts of the model may exhibit ...

14 Incremental formal design verification

Gitanjali M. Swamy, Robert K. Brayton

November 1994 **Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design**

Full text available:  [pdf\(914.23 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Language containment is a method for design verification that involves checking if the behavior of the system to be verified is a subset of the behavior of the specifications (properties or requirements), which it has to meet. If this check fails, language containment returns a subset of "fair" states involved in behavior that the system exhibits but the specification does not. Current techniques for language containment do not take advantage of the fact that the process of desi ...

15 Debugging optimized code without being misled

Max Copperman

May 1994 **ACM Transactions on Programming Languages and Systems (TOPLAS)**

Volume 16 Issue 3

Full text available:  [pdf\(2.57 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Correct optimization can change the behavior of an incorrect program; therefore at times it is necessary to debug optimized code. However, optimizing compilers produce code that impedes source-level debugging. Optimization can cause an inconsistency between where the user expects a breakpoint to be located and the breakpoint's actual location. This article describes a mapping between statements and breakpoint locations that ameliorates this problem. The mapping enables debugger b ...

16 Gross motion planning—a survey

Yong K. Hwang, Narendra Ahuja

September 1992 **ACM Computing Surveys (CSUR)**, Volume 24 Issue 3

Full text available:  [pdf\(6.40 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Motion planning is one of the most important areas of robotics research. The complexity of the motion-planning problem has hindered the development of practical algorithms. This paper surveys the work on gross-motion planning, including motion planners for point robots, rigid robots, and manipulators in stationary, time-varying, constrained, and movable-object environments. The general issues in motion planning are explained. Recent approaches and their performances are briefly described, a ...

Keywords: collision detection, computational geometry, implementation, motion planning, obstacle avoidance, path planning, spatial representation

17 A Fast Implementation of the Minimum Degree Algorithm Using Quotient Graphs

Alan George, Joseph W. H. Liu

September 1980 **ACM Transactions on Mathematical Software (TOMS)**, Volume 6 Issue 3

Full text available:  [pdf\(1.20 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

18 Information flow in hybrid systems

Ruggero Lanotte, Andrea Maggiolo-Schettini, Simone Tini

November 2004 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 3

Issue 4

Full text available:  [pdf\(424.24 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Our aim is to study the information flow problem in hybrid systems, namely systems consisting of a discrete program with an analog environment. Information flows compromise the security of a system because they cause leakage of protected information. In order to tackle information flow in real-life systems, we introduce new classes of hybrid systems that extend the known ones while preserving their properties. Then, we define a logic to specify information flow. By means of some examples, we ...

Keywords: Hybrid systems, information flow**19 Algorithms for computing the static single assignment form**

Gianfranco Bilardi, Keshav Pingali

May 2003 **Journal of the ACM (JACM)**, Volume 50 Issue 3Full text available:  [pdf\(576.44 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Static Single Assignment (SSA) form is a program representation used in many optimizing compilers. The key step in converting a program to SSA form is called ϕ -placement. Many algorithms for ϕ -placement have been proposed in the literature, but the relationships between these algorithms are not well understood. In this article, we propose a framework within which we systematically derive (i) properties of the SSA form and (ii) ϕ -placement algorithms. This framework is based on a n ...

Keywords: Control dependence, optimizing compilers, program optimization, program transformation, static single assignment form**20 Towards dataflow analysis of communicating finite state machines**

W. Peng, S. Purushothaman

June 1989 **Proceedings of the eighth annual ACM Symposium on Principles of distributed computing**Full text available:  [pdf\(1.25 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)